IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants:	RAMAMOORTHY, et al.	Patent Application	
Application No.:	10/690,605	Group Art Unit:	2436
Filed:	October 23, 2003	Examiner:	Hoang, Daniel L.

For: SMART TRANSLATION OF GENERIC CONFIGURATION

APPEAL BRIEF

Table of Contents

I. Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 11445 Compaq Center Drive West, Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. Related Appeals and Interferences

There are no related appeals or interferences known to the Appellants.

III. Status of Claims

Claims 1-4 and 6-18 stand rejected. Claims 1-4 and 6-18 are pending. Claim 5 has been cancelled. This appeal involves Claims 1-4 and 6-18.

IV. Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action mailed August 19, 2010 has not been filed. Therefore, the Clean Copy of the Claims on Appeal in Section VIII of the instant Appeal Brief does reflect the proposed amendments.

V. Summary of Claimed Subject Matter

Independent Claim 1 recites, "A system (100) for implementing a policy in a network," is described, according to various embodiments, at page 4 lines 1-33, and Figure 1. "A plurality of device-agnostic policy implementations (120), in which the device-agnostic policy implementations (120) include non-security policy implementations," is described, according to various embodiments, at page 4 lines 13-24, Figure 1. "A plurality of network devices (140), at least two of said devices (140) being dissimilar, wherein a type of network device (140) associated with a received device-agnostic policy implementation (120) is identified by parsing tags of data from said received device-agnostic policy implementation (120) represented using Extensible Markup Language (XML)," is described, according to various embodiments, at page 4 line 1, page 4 lines 19-20, page 4 lines 27-28, and Figure 1. "A plurality of device translators (130), each device translator (130) corresponding to a respective one of said plurality of network devices (140) and one of said plurality of device-agnostic policy implementations (120), at least two of said device translators (130) being dissimilar, each of said plurality of device translators (130) translating said device-agnostic policy implementation (120) into corresponding device-specific implementations, wherein subsequent additions or maintenance of any of said plurality of said plurality of network device-agnostic policy implementations (120) are provided using device-agnostic files," is described, according to various embodiments, at page 4 lines 21-23, page 4 lines 30-33, and Figure 1.

Independent Claim 10 recites, "A computer-implemented method," which is described, according to various embodiments, at least at page 4 lines 24-33, Figure 2. "Representing (205) a vendor-agnostic configuration using a processor in a computer connected to a computer network," is described, according to various embodiments, at least at page 4 lines 25-26, Figure 2. "Building (210) a translator, using the processor, for a specific policy and vendor, in which the computer network includes a plurality of policies and vendors, the policies including non-security policies," is described, according to various embodiments, at least at page 4 lines 26-27, Figure 2. "Repeating the building for each type of policy and vendor," is

described, according to various embodiments, at least at page 4 lines 30-33. "Identifying (215) a type of device associated with a received vendor-agnostic configuration by parsing tags of data from said received vendor-agnostic configuration representing using Extensible Markup Language (XML), using the processor," is described, according to various embodiments, at least at page 4 lines 27-28, Figure 2. "Loading (220) said translator into memory, using the processor, after identifying said type of device," is described, according to various embodiments, at least at page 4 line 28, Figure 2. "Translating (225) said vendor-agnostic configuration into vendor-specific configuration using said translator," is described, according to various embodiments, at least at page 4 line 29, Figure 2. "Repeating the identifying, loading and translating for each type of policy and vendor; and providing subsequent additions or maintenance of any of said plurality of policies and vendors using device-agnostic files," is described, according to various embodiments, at least at page 4 lines 30-33.

Claim 18 recites, "A computer readable medium containing instructions for implementing a policy in a computer network," which is described, according to various embodiments, at least at page 4 lines 24-33, Figure 2. "Representing (205) a vendor-agnostic configuration," is described, according to various embodiments, at least at page 4 lines 25-26, Figure 2. "Building (210) a translator for a specific policy and a specific vendor, in which the computer network includes a plurality of policies and vendors, the policies including non-security policies" is described, according to various embodiments, at least at page 4 lines 26-27, Figure 2. "Repeating the building for each type of policy and vendor" is described, according to various embodiments, at least at page 4 lines 30-33. "Identifying (215) a type of device associated with a received vendor-agnostic configuration by parsing tags of data from said received vendor-agnostic configuration represented using Extensible Markup Language (XML)," is described, according to various embodiments, at least at page 4 lines 27-28, Figure 2. "Loading (220) said translator after identifying said type of device" is described, according to various embodiments, at least at page 4 line 28, Figure 2. "Translating (225) said received vendor-agnostic configuration into vendor-specific configuration using said translator" is described, according to various embodiments, at least at page 4 line 29, Figure 2. "Repeating the

identifying, loading and translating for each type of policy and vendor; and providing subsequent additions or maintenance of any said plurality of policies and vendors using device-agnostic files," is described, according to various embodiments, at least at page 4 lines 30-33.

VI. Grounds of Rejection to Be Reviewed on Appeal

1. Claims 1-4 and 6-18 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application Publication No. 20020038340 by Whipple et al. (referred to hereinafter as "Whipple") in view of U.S. Patent No. 6,594,823 by Corbin (referred to hereinafter as "Corbin").

VII. Argument

1. Whether Claims 1-4 and 6-18 Are Patentable Over Whipple In View of Corbin Under 35 U.S.C. 103(a)

Appellants have reviewed the asserted art and respectfully submit that the asserted art does not teach or suggest the embodiments recited by the instant Application for at least the following rationale.

"As reiterated by the Supreme Court in KSR, the framework for the objective analysis for determining obviousness under 35 U.S.C. 103 is stated in Graham v. John Deere Co., 383 U.S. 1, 148 USPQ 459 (1966). Obviousness is a question of law based on underlying factual inquiries" including "[a]scertaining the differences between the claimed invention and the prior art" (MPEP 2141(II)). "In determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious" (emphasis in original; MPEP 2141.02(I)).

Appellants note that "[t]he prior art reference (or references when combined) need not teach or suggest all the claim limitations, however, Office personnel must explain why the difference(s) between the prior art and the claimed invention would have been obvious to one of ordinary skill in the art" (emphasis added; MPEP 2141(III)).

Appellants respectfully submit that the instant Office Action fails to explain the differences between the asserted art and Appellants' claimed features, in which the asserted art fails to teach, describe, or suggest features of Appellants' claimed embodiments. Moreover, Appellants respectfully submit that the instant Office Action fails to explain why these differences would have been obvious to one of ordinary skill in the art.

WHIPPLE

This section describes Appellants' understanding of what Whipple teaches. Whipples' title is (emphasis added) "Network Application Program

Interface Facilitating Communication In A Distributed Network Environment."

Whipple states at 0003 lines 1-5,

> As the commercial significance of the Internet has increased, business-to-consumer ("B2C"), business-to business ("B2B"), and other electronic marketplaces have become increasingly prevalent. Each market place typically uses a different set of software applications…

Accordingly, Appellants understand Whipple to teach facilitating communications in a distributed network environment between clients where the communications involve electronic market places such as business-to-consumer ("B2C") or business-to-business ("B2B") with different sets of software applications (see Whipple title, 0003 lines 1-5 quoted herein).

Whipple states at 0023 and 0024 lines 18-21 (emphasis added),

> FIG. 3 illustrates an example request-response protocol associated with the network API supported by hub system 12. In one embodiment, a servlet running in web server 30 acts as a request broker to provide translation between network API (NAPI) and hub API (HAPI) commands. Request broker 50 receives a NAPI request contained within an HTTP request, interprets the format of the wrapper that contains the NAPI request (e.g., Multi-purpose Internet Mail Extensions (MIME)), determines the format in which the one or more parameters of the NAPI request are presented (e.g., XML, EDI, serialized object (JAVA), relational, etc.), selects an appropriate adapter 24 for the NAPI request based on the format of the parameters, and passes the parameters to the selected adapter 24 in their native format using an adapter interface 52. In one embodiment, as is shown, passing the parameters to the selected adapter 24 involves request broker 50 calling a method on adapter interface 52 to get adapter 24 and then calling a method on adapter interface 52 to pass the parameters to adapter 24. Adapter 24 converts the formatted parameters to Java classes or any other representation suitable for processing at hub application server 32, according to the implementation.
>
> …Request broker 50 gives the return value (represented in JAVA classes according to the particular implementation) to the selected adapter 24, which then returns a return value in the same format as the original NAPI request….

Whipple further states at 0016 lines 10-14 (emphasis added), "Hub system 12 includes one or more API adapters 24 suitable for translating the one or more

API formats used by clients 18 to a format appropriate for the hub API, each such format preferably having a corresponding API adapter 24."

Accordingly, Appellants understand Whipple to teach formats that are native to the clients because there are different API formats associated with the clients (see Whipple 0016 lines 10-14 quoted herein). Further, Appellants understand Whipple to teach the format that is appropriate for the hub API is also a native format for at least the reason that Whipple states that each such format preferably has a corresponding API adapter (see Whipple 0016 lines 10-14 quoted herein).

Further, to facilitate the communications, Appellants understand Whipple to teach a request broker that receives a wrapped request, determines the native format of the parameters of the wrapped request where the parameters native format is the requesting client's native format, selects an adapter based on the native format, sends the parameters in the native format to the selected adapter, which then converts the parameters to a recipient's native format (see Whipple 0023, 0016 lines 10-14 quoted herein). Further, Appellants understand Whipple to teach that the process is essentially reversed to provide a return value, where the return value is converted from the recipient's native format to the client's native format (see Whipple 0024 lines 18-21, 0016 lines 10-14 quoted herein).

Appellants understand Whipple to require translating the parameters from one native format into another native format since Whipple's intended purpose is to facilitate communications between clients where the communications involve electronic market places such as business-to-consumer ("B2C") or business-to-business ("B2B") with different sets of software applications (see Whipple title, 0003 lines 1-5 quoted herein).

DIFFERENCES BETWEEN WHIPPLE AND CLAIM 1

Appellants respectfully submit that Whipple determines Whipple's adapter by reading a file wrapper instead of "wherein a type of network device associated with a received device-agnostic policy implantation is identified by parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML)," as recited by independent Claim 1. Further, the Office Action states that Whipple does not teach "parsing tags of data from said received device agnostic policy implementation represented…" as recited. Appellants respectfully agree.

NO MOTIVATION TO COMBINE WHIPPLE WITH ANY OTHER ART

This section describes why there is no motivation to combine Whipple with any other asserted art because Appellants' understand Whipple to teach away from independent Claim 1.

Appellants respectfully submit that "[i]t is improper to combine references where the references teach away from their combination" (emphasis added; MPEP 2145(X)(D)(2); *In re Grasselli*, 713 F.2d 731, 743, 218 USPQ 769, 779 (Fed. Cir. 1983)). Appellants respectfully note that "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention" (emphasis in original; MPEP 2141.02(VI); *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984)). Further, "[a] reference will teach away if it suggests that the line of development flowing from the reference's disclosures is unlikely to be productive of the result sought by the applicant. *In re Gurley*, 31 USPQ2d 1130 (Fed. Cir. 1994)."

Appellants respectfully submit that Whipple does not teach or suggest "a plurality of device-agnostic policy implementations…a plurality of device translators, each device translator corresponding to a respective one of said plurality of network devices…each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," as recited by independent Claim 1.

First, as discussed herein, Appellants understand Whipple to teach translating the parameters from one native format into another native format (see Whipple 0023, 0016 lines 10-14 quoted herein) instead of "device agnostic policy implementations… translating said device-agnostic policy implementation into corresponding device-specific implementations," as recited by independent Claim 1.

Further, Appellants respectfully submit that Whipple teaches away from "…translating said device-agnostic policy implementation into corresponding device-specific implementations." For example, as discussed herein, Appellants understand Whipple to require translating the parameters from one native format into another native format in order to achieve Whipple's intended purpose is to facilitate communications between clients where the communications involve electronic market places such as business-to-consumer ("B2C") or business-to-business ("B2B") with different sets of software applications (see Whipple title, 0003 lines 1-5 quoted herein). Appellants respectfully submit that requiring translation of parameters from one native format into another native format teaches away from "…translating said device-agnostic policy implementation into corresponding device-specific implementations."

Second, Appellants understand Whipple to teach that there is an adapter for each requesting client's native format since Whipple teaches selecting an adapter based on the requesting client's native format where the adapter converts parameters from one native format into another native format (see Whipple 0023, 0016 lines 10-14 quoted herein). Appellants respectfully submit that an adaptor for each requesting client's native format where the adapter converts parameters from one native format into another native format does not teach or suggest "each device translator corresponding to a respective one of said plurality of network devices…each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," as recited by independent Claim 1. Further, Appellants respectfully submit that selecting an adapter based on the requesting

client's native format where the adapter converts parameters from one native format into another native format (see Whipple 0023, 0016 lines 10-14 quoted herein) teaches away from "each device translator corresponding to a respective one of said plurality of network devices…each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," as recited by independent Claim 1.

Appellants respectfully submit that since Whipple teaches away from Claim 1, there is no motivation to combine Whipple with any other asserted art, such as Corbin.

Further, since Appellants do not understand Whipple to teach or suggest "a plurality of device-agnostic policy implementations……a plurality of device translators, each device translator corresponding to a respective one of said plurality of network devices …each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," as recited by independent Claim 1, Appellants respectfully submit that Whipple cannot teach or suggest any of the other features recited by independent Claim 1. Therefore, Appellants respectfully submit that Whipple does not teach or suggest,

> a plurality of device-agnostic policy implementations, in which the device-agnostic policy implementations include non-security policy implementations;
>
> a plurality of network devices, at least two of said devices being dissimilar, wherein a type of network device associated with a received device-agnostic policy implementation is identified by parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML); and
>
> a plurality of device translators, each device translator corresponding to a respective one of said plurality of network devices, at least two of said device translators being dissimilar, each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations, wherein subsequent additions or maintenance of any of said plurality of said plurality of network device-agnostic policy implementations are provided using device-agnostic files,

as recited by independent Claim 1.

CORBIN

This section describes Appellants' understanding of what Corbin teaches. Appellants understand Corbin to teach using tags to create a data structure that is generic with respect to different programming languages to avoid writing the data structure in each applicable language (title, abstract, Col. 1 lines 15-28, Col. 4 lines 13-16).

CORBIN DOES NOT REMEDY THE DEFICIENCIES OF WHIPPLE

Appellants respectfully submit that Corbin does not remedy the deficiencies in Whipple. For example, as discussed herein, Appellants do not understand Whipple to teach or suggest "a plurality of device-agnostic policy implementations… wherein a type of network device associated with a received device-agnostic policy implantation is identified by parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML)…a plurality of device translators…each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," as recited by independent Claim 1.

For example, Corbin title is "Method and System for Representing a High-Level Programming Language Data Structure In A Mark-Up Language." Corbin states at Col. 1 lines 15-28,

> There are currently a myriad of high-level programming languages available for a programmer to use. Some of the more popular ones include Java, Java Script and C++. Each high-level language has its own way of defining data structures, such as arrays, integers, strings, and the like. However, a data structure written in the source code of one language generally cannot be compiled by a compiler of a different language. Thus, if a programmer is working on a system in which two or more different programming languages are being used, but all require access to the same data structure, then he or she is forced to write the data structure in each applicable language. This is particular cumbersome when the data structure needs to be changed during the debugging process.

Corbin also states at Col. 4 lines 13-16, "For example, a data structure may be written once in XML and translated into several high-level programming languages, such as C++, Java, ADA, VisualBasic, and Pascal."

Accordingly, Appellants understand Corbin to teach using tags to create a data structure that is generic with respect to different programming languages (see Corbin title, Col. 1 lines 15-28, Col. 4 lines 13-16 quoted herein). Appellants respectfully submit that a data structure that is generic with respect to different programming languages does not remedy the deficiencies in Whipple.

Second, Appellants respectfully submit that since Whipple teaches away from "device-agnostic policy implementations," "translating said device-agnostic policy implementation into corresponding device-specific implementations," and "each device translator corresponding to a respective one of said plurality of network devices…each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," as discussed herein, there is no motivation to combine Corbin with Whipple.

RESPONSE TO ARGUMENTS SECTION

The Office Action states at lines 6-9 of paragraph 1 on page 2, "…It is examiner's understanding of applicant's invention that applicant is claiming a system wherein code is written that is non-vendor specific. A translator is then built that translate this code into a vendor specific code. Multiple translators are built for each type of vendor specific code (applicant's specification, paragraph 20). Appellants reiterate that Claim 1 recites,

> a plurality of device-agnostic policy implementations, in which the device-agnostic policy implementations include non-security policy implementations;
>
> a plurality of network devices, at least two of said devices being dissimilar, wherein a type of network device associated with a received device-agnostic policy implementation is identified by parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML); and

> a plurality of device translators, each device translator corresponding to a respective one of said plurality of network devices, at least two of said device translators being dissimilar, each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations, wherein subsequent additions or maintenance of any of said plurality of said plurality of network device-agnostic policy implementations are provided using device-agnostic files.

The Office Action states at lines 10-11 of paragraph 1 on page 2, "Examiner contends that Whipple teaches the same embodiments. Whipple takes code of a specific native format and translates it into an internal format." Appellants respectfully submit that Whipple does not teach translating code from one format into another format. For example, Whipple states at 0026 lines 1-3 (emphasis added), "As the above description makes clear, communication between request broker 50 and client 18 involves a request-response protocol." Further, Whipple states at 0006 lines 3-10 (emphasis added),

> Certain embodiments of the invention may provide a mechanism for describing and executing an application program interface (API in a distributed network environment for describing and executing an application program interface (API) in a distributed network environment such as the Internet, including providing the ability to receive and return values form API calls and to return exception and deprecation notices when appropriate. Certain embodiments of the invention may allow disparate remote clients to interact…

Accordingly, Appellants understand Whipple to teach enabling different types of clients to communicate with each other that involves wrapping requests and converting formats of the requests and responses (see Whipple 0026 lines 1-3, 0006 lines 3-10, 0016 lines 10-14, 0023 and 0024 lines 18-21 quoted herein) instead of teaching the format of code as the Office Action asserts at lines 10-11 of paragraph 1 on page 2.

The Office Action states at lines 14-17 of paragraph 1 on page 2, "The fact that the internal format code is generated from a native format is a moot point. Applicant's claim does not specify the format from which the device agnostic code is generated from. As such, Whipple clearly teaches translating code from

a non vendor specific format into a vendor specific format." Appellants respectfully note that the Office Action states in the first paragraph on page 4,

> [see paragraph 16, wherein communication from clients are translated into an appropriate internal format for the HUB API. Examiner views the internal formats from that are appropriate for the HUB API as device-agnostic policy implementations. Said formats are device agnostic because they are not specific to the clients but rather formatted from the clients' specific formats into formats that are suitable for the HUB.]

Therefore, Appellants respectfully submit that the Office Action is comparing the formats of Whipple's communications to "implementations" as recited by Claim 1.

Further, Appellants respectfully agree with the Office Action's assertion in the first paragraph on page 4 that Whipple is directed to communications between the clients and the HUB involve translating formats of the respective communications. Therefore, the Office Action has admitted in the first paragraph on page 4 that Whipple is not directed to translating code of a specific native format and into an internal format as the Office Action contended at lines 10-11 of paragraph 1 on page 2.

The Office Action states on page 3 lines 1-2, "Applicant is arguing that Whipple translates from a device specific policy into another device specific policy." Appellants respectfully submit that this is a mischaracterization of Appellants' statements. Appellants have not argued that Whipple teaches "policies" or "implementations" as recited. Instead, Appellants have stated that Whipple is directed to wrapping communications and translating formats of communications.

The Office Action states on page 3 lines 2-5, "Whipple has an adapter which translates a native format into an internal format. The internal format can then be translated into any of a plurality of native formats using the right adapter. As explained below, the internal format is viewed as device agnostic because it can be translated into any native format." Whipple states at 0016 lines 10-14 (emphasis added), "Hub system 12 includes one or more API adapters 24

suitable for translating the one or more API formats used by clients 18 to a format appropriate for the hub API, each such format preferably having a corresponding API adapter 24." Accordingly, Appellants understand Whipple to teach the format that is appropriate for the hub API is also a native format for at least the reason Whipple states that each such format preferably has a corresponding API adapter (see Whipple 0016 lines 10-14 quoted herein).

SUMMARY

Therefore, Appellants respectfully submit that Claim 1 is patentable over the combination of Whipple and Corbin in that neither Whipple or Corbin teach or suggest "a plurality of device-agnostic policy implementations… wherein a type of network device associated with a received device-agnostic policy implantation is identified by parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML)…a plurality of device translators, each device translator corresponding to a respective one of said plurality of network devices …each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," and Whipple teaches away from "device-agnostic policy implementations," "translating said device-agnostic policy implementation into corresponding device-specific implementations," and "each device translator corresponding to a respective one of said plurality of network devices…each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations," among other things, as recited by independent Claim 1.

For similar reasons, Appellants respectfully submit that independent Claims 10 and 18 are also patentable over the Whipple Corbin combination.

Claims 2-4 and 6-9 depend on independent Claim 1. Claims 11-17 depend on independent Claim 10. These dependent claims include all of the features of their respective independent claims. Therefore, these dependent claims should be patentable for at least the reasons that their respective independent claims should be patentable.

Conclusion

Appellants believe that pending Claims 1-4 and 6-18 are patentable over Whipple and Corbin. As such, Appellants submit that Claims 1-4 and 6-18 are patentable over the asserted art.

Appellants respectfully request that the rejection of Claims 1-4 and 6-18 be reversed. The Appellants wish to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellants' undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,
Wagner Blecher LLP

Dated: 07/12/2011

/John P. Wagner, Jr./

John P. Wagner, Jr.
Registration No.: 35,398

Wagner Blecher LLP
Westridge Business Park
123 Westridge Drive
Watsonville, CA 95076

Phone: (408) 377-0500
Facsimile: (831) 722-2350

VIII. Appendix - Clean Copy of Claims on Appeal

1. A system for implementing a policy in a network, said system comprising:

a plurality of device-agnostic policy implementations, in which the device-agnostic policy implementations include non-security policy implementations;

a plurality of network devices, at least two of said devices being dissimilar, wherein a type of network device associated with a received device-agnostic policy implementation is identified by parsing tags of data from said received device-agnostic policy implementation represented using Extensible Markup Language (XML); and

a plurality of device translators, each device translator corresponding to a respective one of said plurality of network devices and one of said plurality of device-agnostic policy implementations, at least two of said device translators being dissimilar, each of said plurality of device translators translating said device-agnostic policy implementation into corresponding device-specific implementations, wherein subsequent additions or maintenance of any of said plurality of said plurality of network device-agnostic policy implementations are provided using device-agnostic files.

2. The system according to claim 1, wherein said device-agnostic policy implementation is selected from the group consisting of firewall, Virtual Private Network, Java 2 Enterprise Edition Application, and custom operating system.

3. The system according to claim 1, wherein said device-agnostic policy implementation implements a policy selected from the group consisting of access control, quality of service, backup, and availability.

4. The system according to claim 1, wherein said device translators are represented by Extensible Stylesheet Language (XSL) code.

6. The system according to claim 3, wherein said policy is represented by Extensible Markup Language (XML) code.

7. The system according to claim 1, wherein the device-specific implementation is represented by Command Line Interface (CLI) code.

8. The system according to claim 1, wherein the device-specific implementation is represented by Application Programming Interface (API) code.

9. The system according to claim 1, wherein the device-specific implementation is represented by Java code.

10. A computer-implemented method comprising:

representing a vendor-agnostic configuration using a processor in a computer connected to a computer network;

building a translator, using the processor, for a specific policy and vendor, in which the computer network includes a plurality of policies and vendors, the policies including non-security policies;

repeating the building for each type of policy and vendor;

identifying a type of device associated with a received vendor-agnostic configuration by parsing tags of data from said received vendor-agnostic configuration representing using Extensible Markup Language (XML), using the processor;

loading said translator into memory, using the processor, after identifying said type of device;

translating said vendor-agnostic configuration into vendor-specific configuration using said translator; and

repeating the identifying, loading and translating for each type of policy and vendor; and

providing subsequent additions or maintenance of any of said plurality of policies and vendors using device-agnostic files.

11. The method according to claim 10, wherein said vendor-agnostic configuration is represented by Extensible Markup Language (XML) code.

12. The method according to claim 10, wherein said translator is represented by Extensible Stylesheet Language (XSL) code.

13. The system according to claim 10, wherein said specific policy is selected from the group consisting of firewall, Virtual Private Network, Java 2 Enterprise Edition Application, and custom operating system.

14. The system according to claim 10, wherein said specific policy is selected from the group consisting of access control, quality of service, backup, and availability.

15. The system according to claim 10, wherein the vendor-specific configuration is represented by Command Line Interface (CLI) code.

16. The system according to claim 10, wherein the vendor-specific configuration is represented by Application Programming Interface (API) code.

17. The system according to claim 10, wherein the vendor-specific configuration is represented by Java code.

18. A computer readable medium containing instructions for implementing a policy in a computer network, said instructions comprising:

representing a vendor-agnostic configuration;

building a translator for a specific policy and a specific vendor, in which the computer network includes a plurality of policies and vendors, the policies including non-security policies;

repeating the building for each type of policy and vendor;

identifying a type of device associated with a received vendor-agnostic configuration by parsing tags of data from said received vendor-agnostic configuration represented using Extensible Markup Language (XML);

loading said translator after identifying said type of device;

translating said received vendor-agnostic configuration into vendor-specific configuration using said translator;

repeating the identifying, loading and translating for each type of policy and vendor; and

providing subsequent additions or maintenance of any said plurality of policies and vendors using device-agnostic files.

IX. Evidence Appendix

No evidence is herein appended.

X. Related Proceedings Appendix

No related proceedings.